

Réalisation d'un compteur numérique avec une carte Arduino UNO

Pour simuler un compteur numérique qui compte dans l'ordre croissant de 0 à 9, nous vous proposons un nouvel atelier de découverte pour vous initier avec les cartes programmables **UNO** de la famille **Arduino**. Pour cela, nous allons utiliser un **afficheur 7 segments** (à base de diode LED) et une **carte Arduino UNO Rev3. R3**

Schéma proposé d'un compteur numérique de 0 à 9

Pour réaliser le compteur numérique, qui va nous réaliser le comptage de 0 à 9, à base d'un afficheur à 7 segments, nous allons câbler le schéma ci-dessous à l'aide d'une plaque d'essai, des résistances et d'un afficheur à 7 segments :

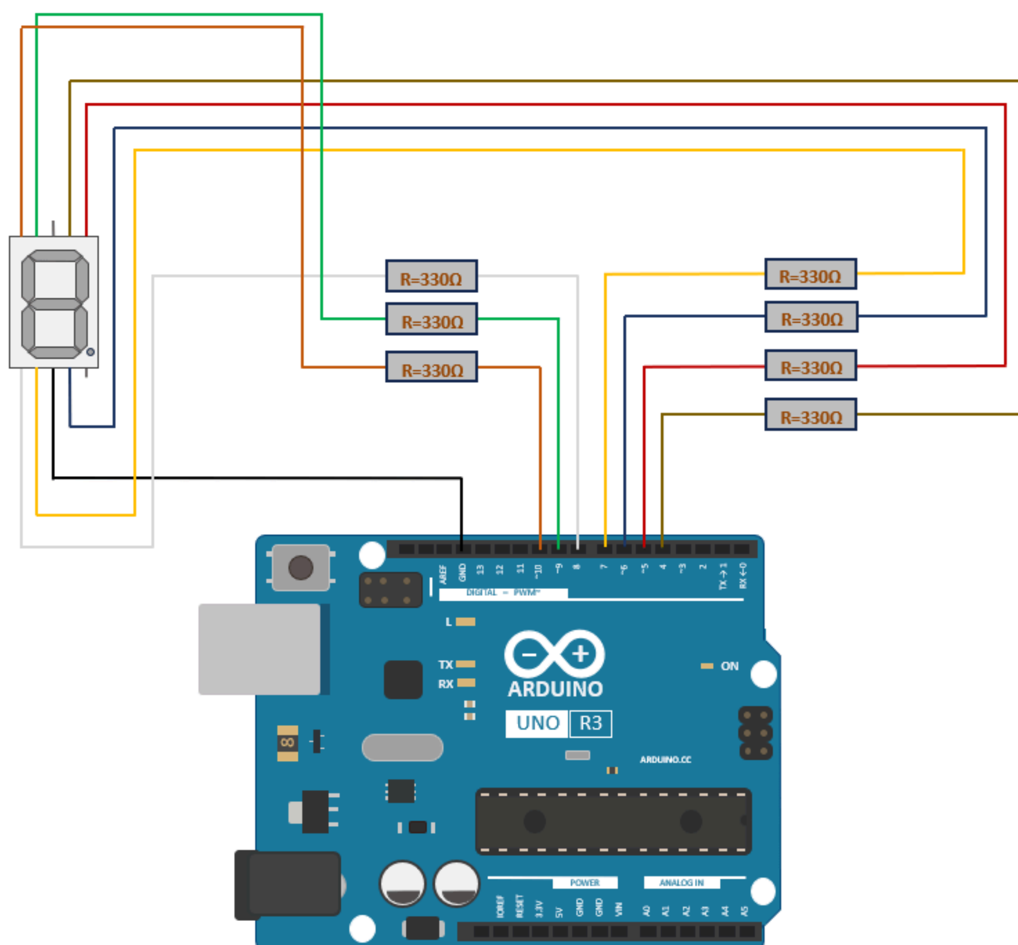


Schéma d'un compteur numérique à base d'une carte Arduino UNO
et d'un afficheur à 7 segments

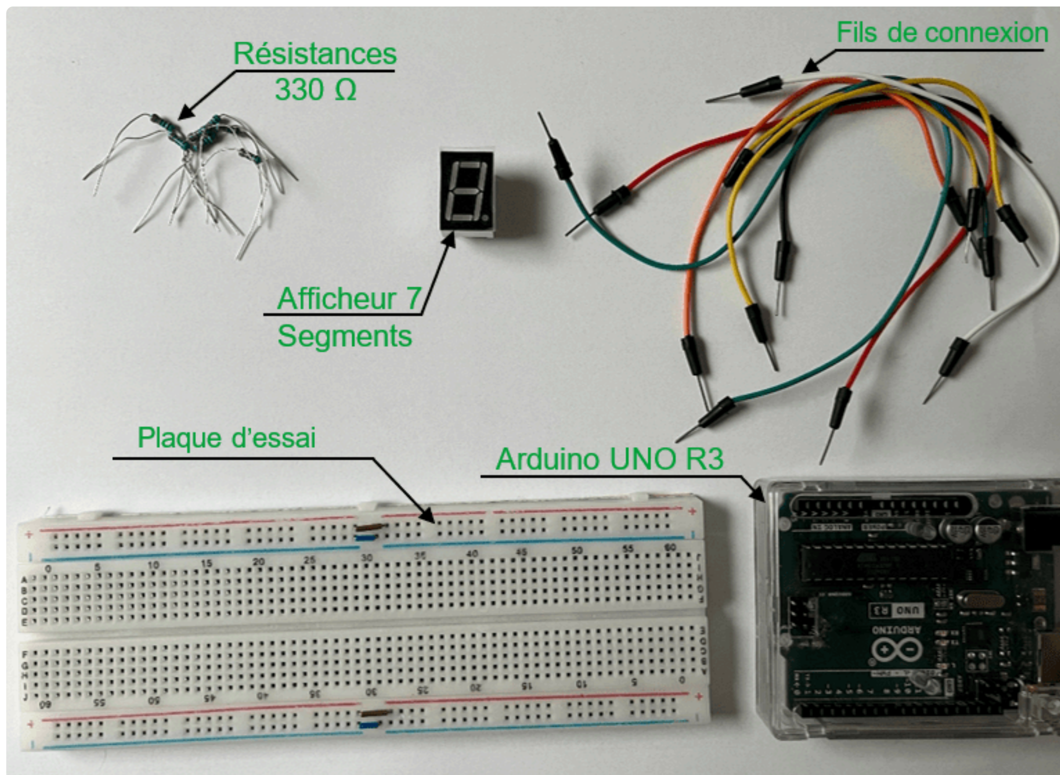
Comme nous pouvons le constater sur notre schéma de montage, nous aurons besoin des composants suivants pour réaliser cette activité :

- Une plaque d'essai,
- Un **afficheur à 7 segments** de couleur rouge.
- Sept **résistances** d'une valeur chacune de **330 Ohm**,
- Des fils de connexion,
- Et une carte **Arduino UNO R3**

Après avoir préparé nos différents composants nécessaires, nous pouvons commencer notre câblage. L'image ci-dessous nous détaille ces composants une par une requis pour cette activité pratique :



<



Comme vous l'avez remarqué, pour protéger les différentes **diodes LED** dans notre afficheur à 7 segments, nous avons utilisé une résistance par segment d'une valeur de **330Ω** en série.



Code couleur d'une résistance électrique

Si vous avez besoin de vérifier la valeur ohmique de vos résistances, vous pouvez utiliser la calculatrice disponible dans notre section les outils proposés par electro-robot :

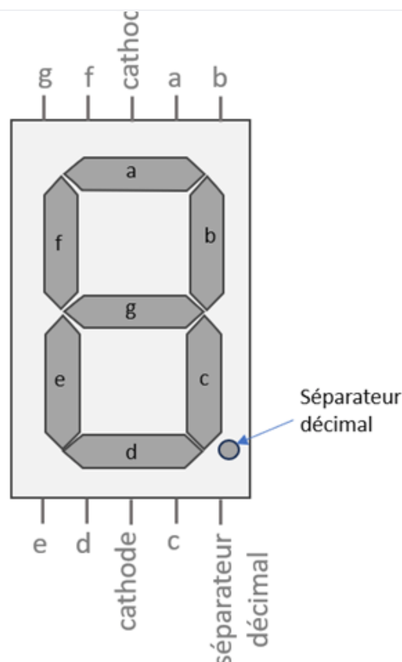
- **Calculatrice de codes couleur des résistances**

Une résistance d'une valeur de **330Ω** dispose des bandes de couleurs suivantes de gauche à droite : **orange** pour la première bande, **orange** pour la deuxième bande, **noir** pour la troisième bande, **noir** pour la quatrième bande et la couleur **marron** pour la dernière.

Comment fonctionne un afficheur à 7 segments ?

Un afficheur à **7 segments** est un afficheur composé de plusieurs **diodes LED** disposées en forme de 8 avec un point de digit. Un afficheur à 7 segments est généralement utilisé pour afficher des chiffres en format numérique mais aussi quelques chiffres tels que la lettre A, B, C, D, E, F, H, etc. L'image ci-dessous nous détaille comment ces diodes LED sont disposé :





Ce composant on le retrouve assez souvent dans les montres, les calculatrices, les compteurs, les horloges numériques, les segments d'affichage pour certains messages, etc.

Pour un afficheur **à 7 segments** fonctionne, nous aurons à alimenter donc certaines diodes LED des segments requis pour afficher le chiffre ou la lettre voulue.

Par exemple, pour afficher le chiffre **1**, nous aurons à alimenter simultanément les deux segments **b** et **c** de notre afficheur. Pour cela, nous allons mettre la cathode montée en série avec une résistance **330Ω** à la masse et les deux broches b et c à la tension Vcc (en général, en fonction de la technologie utilisée, cette tension d'alimentation est de l'ordre de 2V).

Câblage du Montage électrique et codage

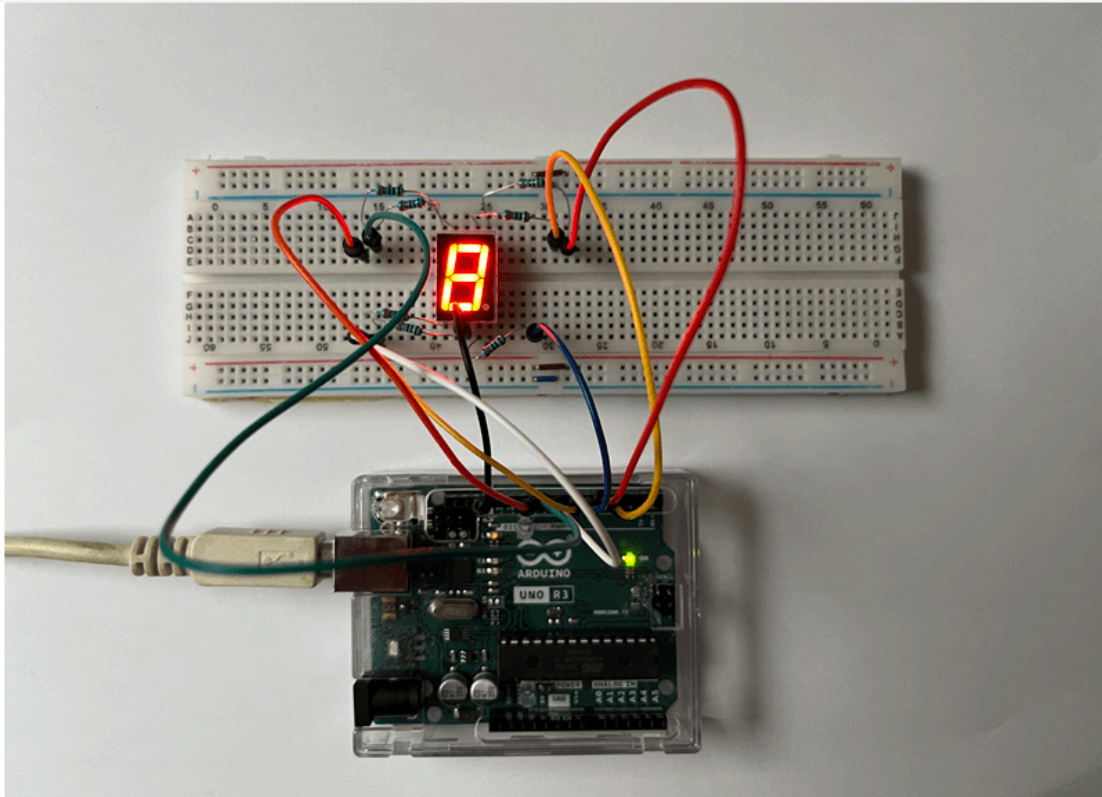
Montage du schéma électrique proposé

Comme le schéma nous l'indique dans la photo ci-dessus (le schéma de montage fourni tout au début de cette activité), nous allons utiliser 7 pins de notre carte **Arduino UNO** telle que:

- **Pin 4** : pour alimenter la diode LED rouge du segment **a** de notre afficheur,
- **Pin 5** : pour alimenter la diode LED rouge du segment **b**,
- **Pin 6** : pour alimenter la diode LED rouge du segment **c**,
- **Pin 7** : pour alimenter la diode LED rouge du segment **d**,
- **Pin 8** : pour alimenter la diode LED rouge du segment **e**,
- **Pin 9** : pour alimenter la diode LED rouge du segment **f**,
- **Pin 10** : pour alimenter la diode LED rouge du segment **g**.

Une fois que nous avons réalisé le câblage de notre schéma proposé, notre montage devra ressembler à la photo ci-dessous :





Attention : Nous vous rappelons qu'il est recommandé de faire attention lorsque vous manipulez votre carte Arduino. Si vous ne disposez pas encore des compétences avancées en électronique, n'utilisez jamais une source d'énergie externe couplée avec votre carte. Cela pourra vous endommager votre carte **Arduino UNO**.

Dans cette activité proposée, nous n'utilisons uniquement qu'une **carte Arduino** sans aucune autre **source d'alimentation externe**. Comme vous l'avez remarqué, la GND (le point de référence 0V) dans notre circuit et celle de notre carte déployée.

Codage, compilation et téléversement

Afin de commencer à écriture de notre code en **langue C++** sous **Arduino IDE,** nous allons définir un scénario simple pour le bon fonctionnement pour notre appliation **compteur dans l'ordre croissant numérique de 0 à 9** à base un **afficheur de 7 segments** tel que:

1. Afficher 0, puis attendre 2 secondes,
2. Afficher 1, puis attendre 2 secondes,
3. Afficher 2, puis attendre 2 secondes,
4. Afficher 3, puis attendre 2 secondes,
5. Afficher 4, puis attendre 2 secondes,
6. Afficher 5, puis attendre 2 secondes,
7. Afficher 6, puis attendre 2 secondes,
8. Afficher 7, puis attendre 2 secondes,
9. Afficher 8, puis attendre 2 secondes,
10. Afficher 9, puis attendre 2 secondes → aller à l'étape 1.

Pour cela, nous allons mettre dans une matrice à 2 dimensions (une matrice est un tableau organisé en lignes et colonnes. Elle peut contenir au minimum 2 dimensions ou plusieurs) les valeurs exprimées en 0 ou 1 qui nous indiquent respectivement éteindre ou allumer les segments de notre afficheur à 7 segments. Par exemple, pour afficher le chiffre 0, nous aurons la série 1,1,1,1,1,1,0 et pour afficher le chiffre 1, nous aurons la suite suivante 0,1,1,0,0,0,0. Ce qui nous donne la matrice suivante :

```
byte chiffres[10][7] = {
  // a, b, c, d, e, f, g
  {1,1,1,1,1,1,0}, // 0
  {0,1,1,0,0,0,0}, // 1
  {1,1,0,1,1,0,1}, // 2
  {1,1,1,1,0,0,1}, // 3
  {0,1,1,0,0,1,1}, // 4
  {1,0,1,1,0,1,1}, // 5
  {1,0,1,1,1,1,1}, // 6
  {1,1,1,0,0,0,0}, // 7
```



```
};
```

Ensuite, nous envoyons bloc par bloc via la fonction **digitalWrite** les différentes valeurs d'une ligne (que nous pouvons aussi appelé vecteur) vers les sorties respectives de notre **carte Arduino** espacés de 2 secondes. Pour cela, nous allons écrire une fonction **afficherChiffre** qui lit les valeurs d'une entrée (soir notre vecteur en question) vers les pins de notre carte numérotés de 4 à 10.

```
// Fonction d'affichage d'un chiffre
void afficherChiffre(int n) {
  for (int i = 0; i < 8; i++) {
    digitalWrite(segments[i], chiffres[n][i]);
  }
}
```

Avec le tableau segment qui représente le tableau des broches dans l'ordre de notre afficheur 7 segments. Sa définition est comme suit :

```
// Tableau des broches dans l'ordre
int segments[] = {a, b, c, d, e, f, g};
```

Maintenant sous **Arduino IDE,** si nous écrivons notre code, nous aurons le code suivant :

```
// Définition des broches reliées à chaque segment de notre afficheur 7 segments
const int a = 4;
const int b = 5;
const int c = 6;
const int d = 7;
const int e = 8;
const int f = 9;
const int g = 10;

// Définition du tableau des segments de l'afficheur dans l'ordre alphabétique
int segments[] = {a, b, c, d, e, f, g};

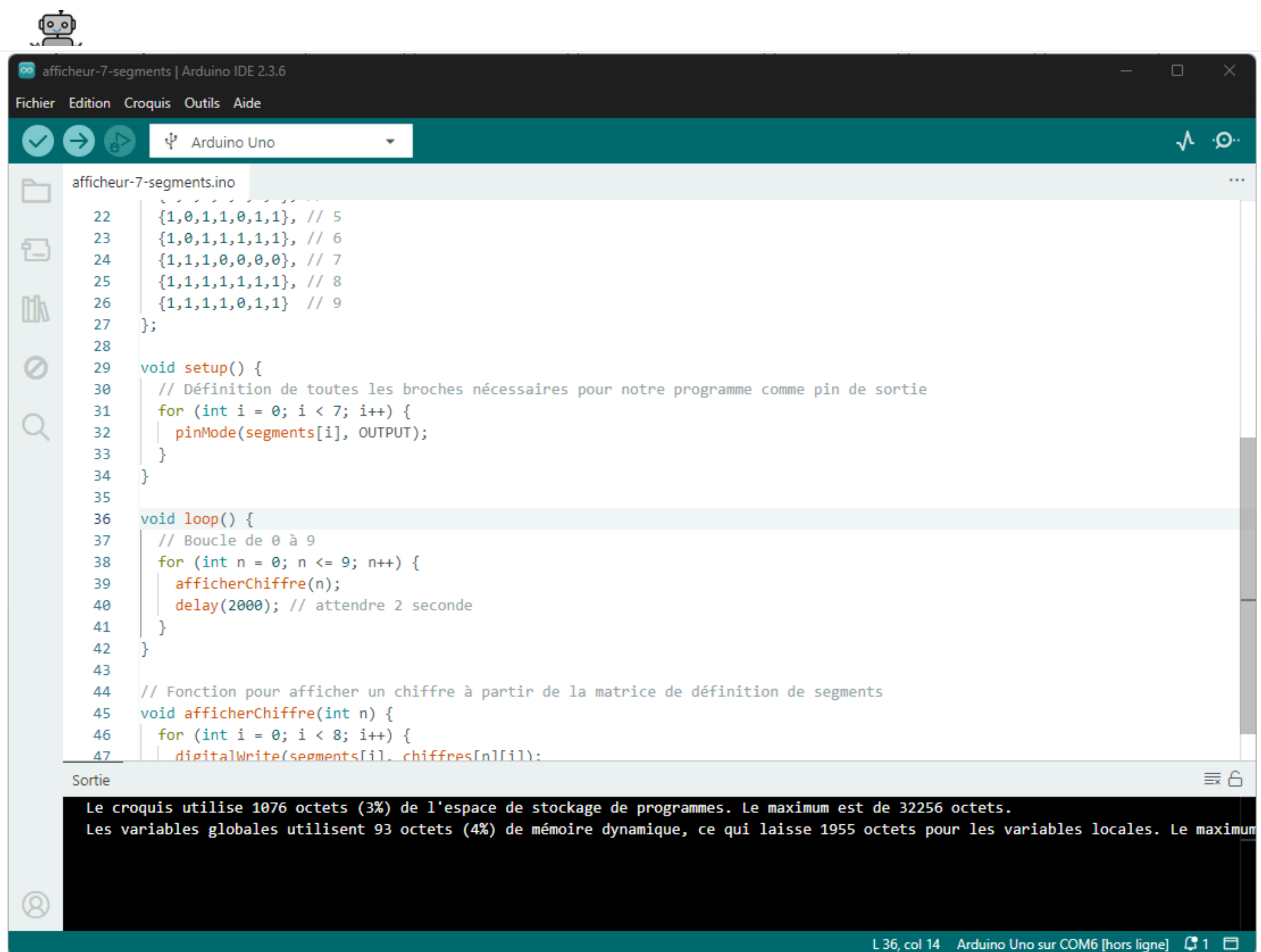
// Déclaration de la matrice des états de chaque segment pour chaque affichage de chaque chiffre de 0 à 9
// Pour cela, nous définissons 1 pour un segment allumé et 0 pour un segment éteint
byte chiffres[10][7] = {
  // a, b, c, d, e, f, g, dp
  {1,1,1,1,1,1,0}, // 0
  {0,1,1,0,0,0,0}, // 1
  {1,1,0,1,1,0,1}, // 2
  {1,1,1,1,0,0,1}, // 3
  {0,1,1,0,0,1,1}, // 4
  {1,0,1,1,0,1,1}, // 5
  {1,0,1,1,1,1,1}, // 6
  {1,1,1,0,0,0,0}, // 7
  {1,1,1,1,1,1,1}, // 8
  {1,1,1,1,0,1,1}  // 9
};

void setup() {
  // Définition de toutes les broches nécessaires pour notre programme comme pin de sortie
  for (int i = 0; i < 7; i++) {
    pinMode(segments[i], OUTPUT);
  }
}

void loop() {
  // Boucle de 0 à 9
  for (int n = 0; n <= 9; n++) {
    afficherChiffre(n);
    delay(2000); // attendre 2 seconde
  }
}

// Fonction pour afficher un chiffre à partir de la matrice de définition de segments
void afficherChiffre(int n) {
  for (int i = 0; i < 8; i++) {
    digitalWrite(segments[i], chiffres[n][i]);
  }
}
```

L'image ci-dessous nous présente notre code écrit sous **Arduino IDE** :



Une fois que vous avez terminé d'écrire votre code sous **Arduino IDE**, vous pouvez maintenant connecter votre carte **Arduino UNO** à votre ordinateur via un câble **USB-B,** lancer la compilation et ensuite le téléversement.



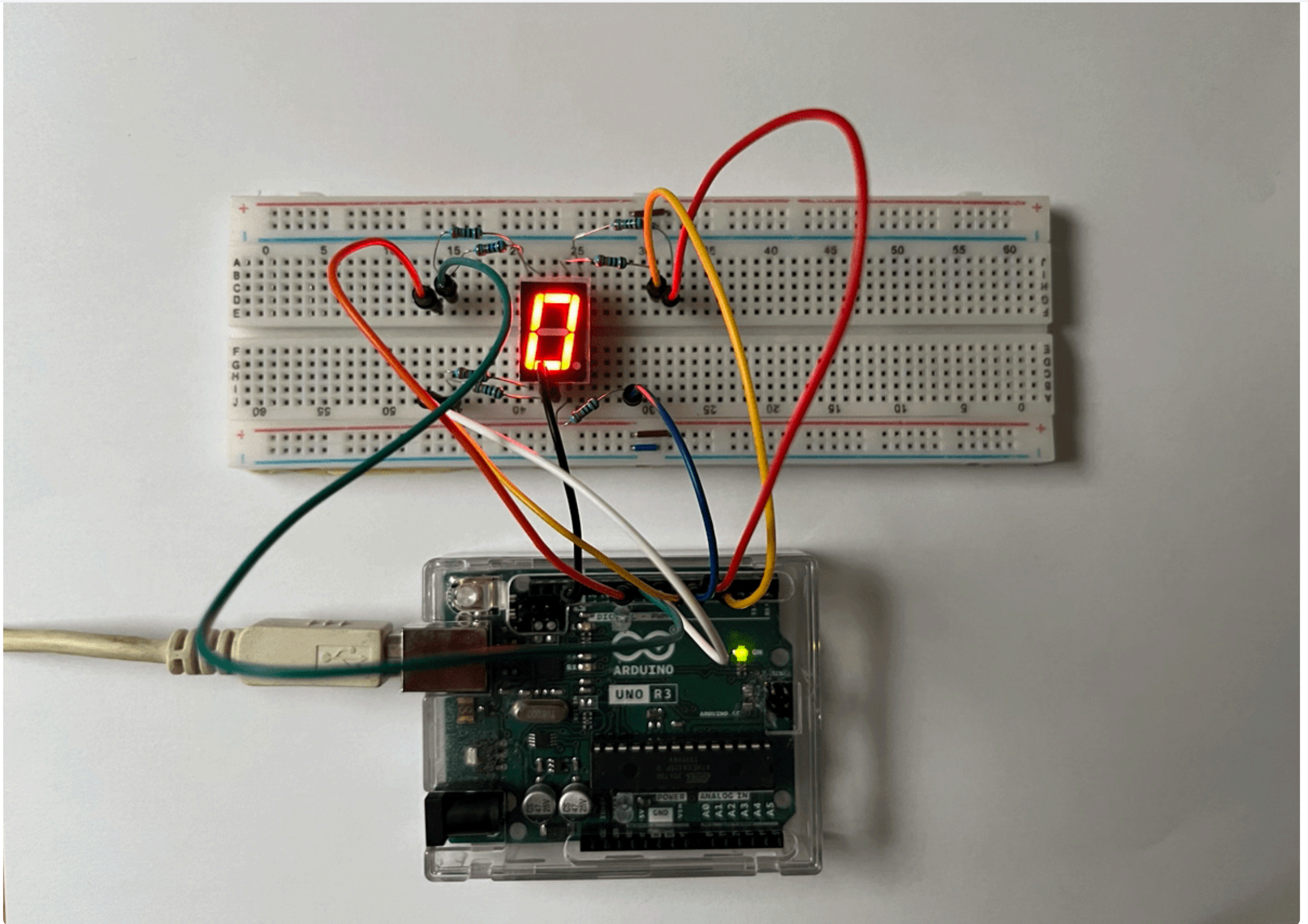
Si lors de la compilation, vous aurez une erreur ou plusieurs, nous vous recommandons de vérifier dans un premier temps votre code si vous l'avez correctement écrit (la fenêtre sortie vous indique les erreurs que vous rencontrez lors de la compilation de votre programme).
Dans le cas que l'erreur persiste lors de la compilation ou du téléversement de votre code vers votre carte, nous vous recommandons de vérifier sa configuration, son port sous **Arduino IDE** (vous devez vérifier cela dans le gestionnaire de cartes accessible sous **Arduino IDE** à partir du raccourci **Ctrl+Maj+B**)

Exécution et conclusion

Enfin, une fois notre montage est fait, notre code est écrit, compilé et téléversé vers notre carte, vous remarquez parfaitement que notre afficheur nous affiche le résultat voulu :

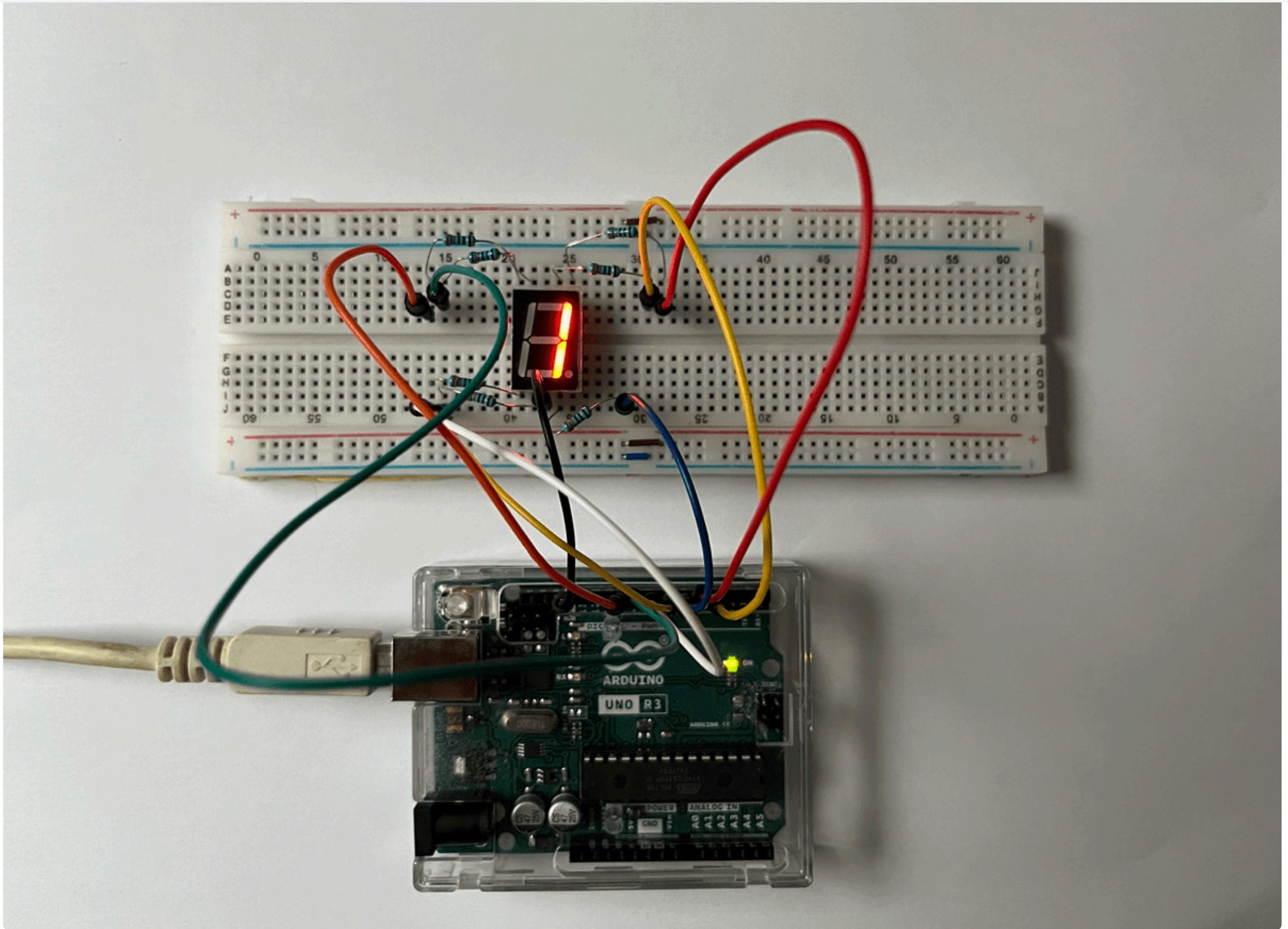
- Le chiffre **0** s'affiche sur notre afficheur 7 segments pendant 2 secondes. L'image ci-dessous vous le montre :





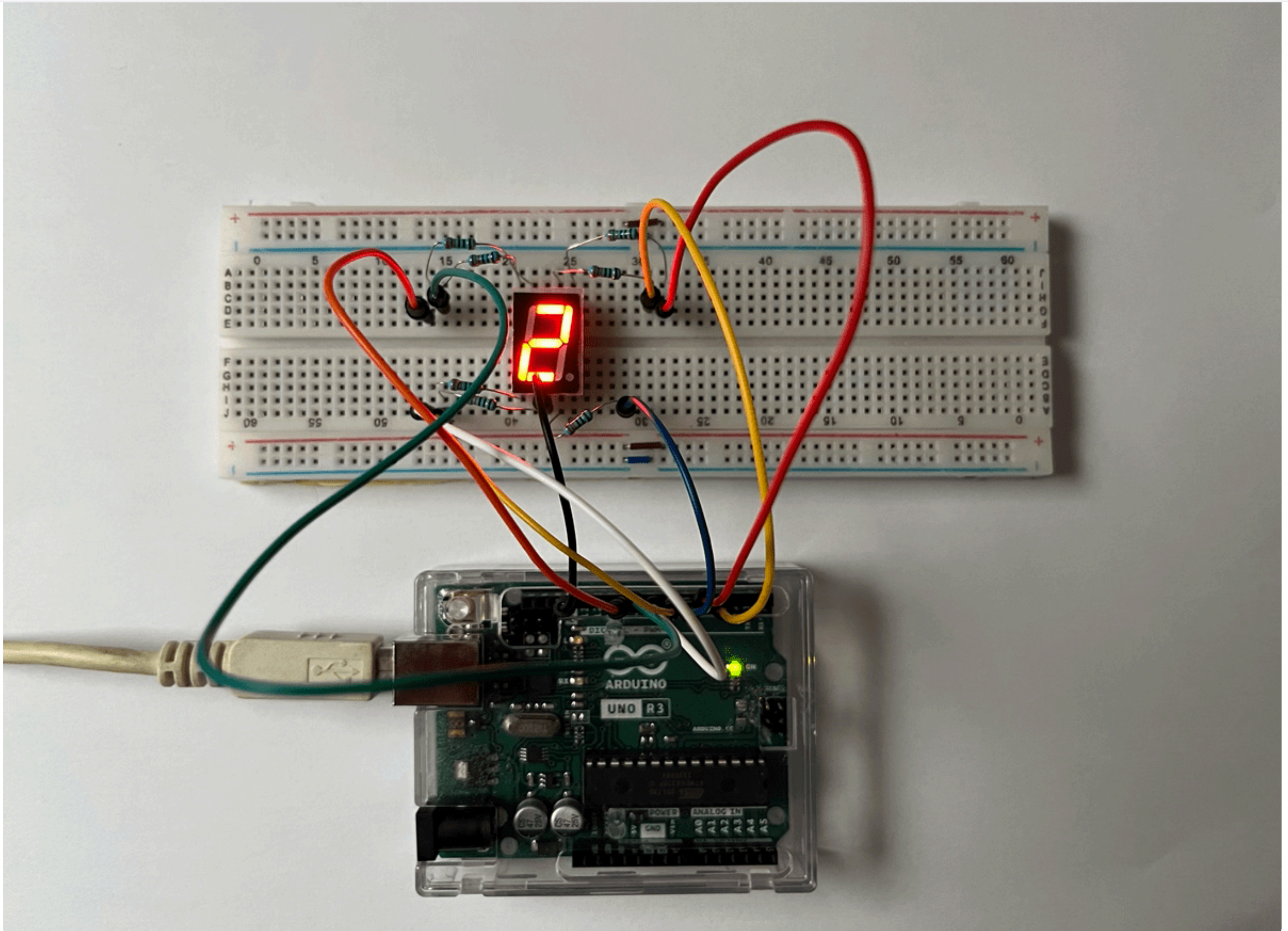
- Le chiffre **1** s'affiche sur notre afficheur 7 segments pendant 2 secondes :





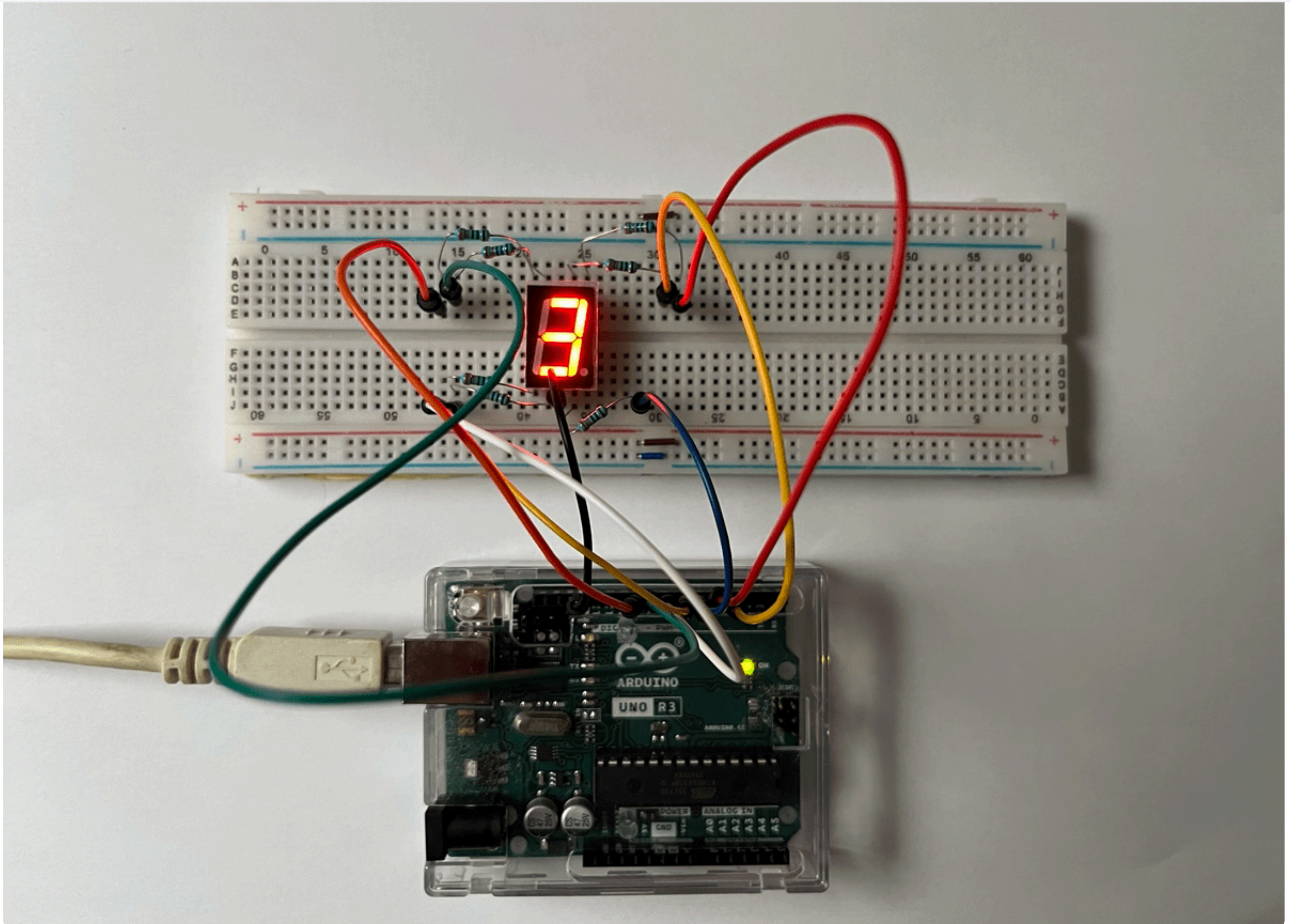
- Le chiffre **2** s'affiche sur notre afficheur 7 segments pendant 2 secondes :





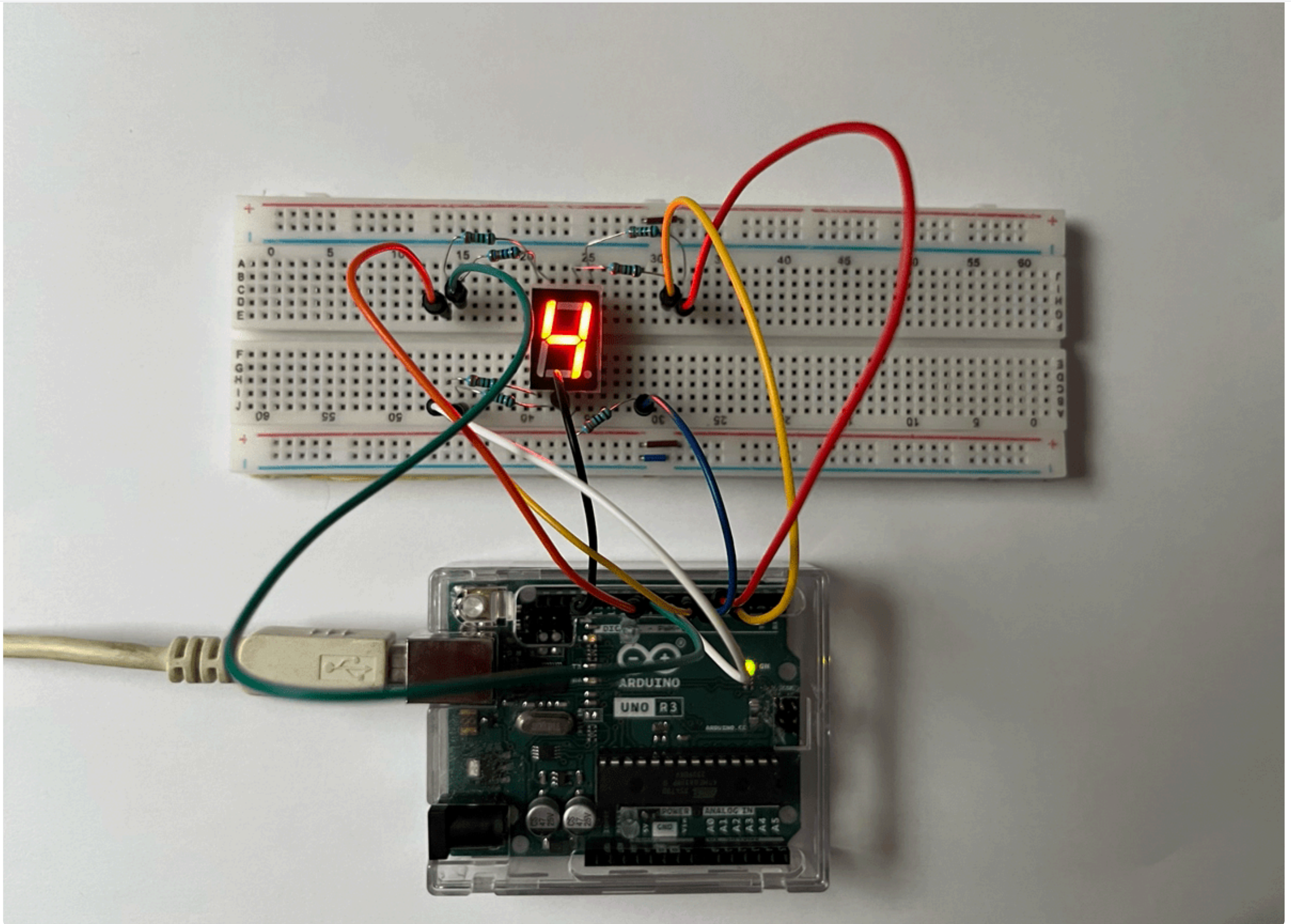
- Le chiffre **3** s'affiche sur notre afficheur 7 segments pendant 2 secondes :





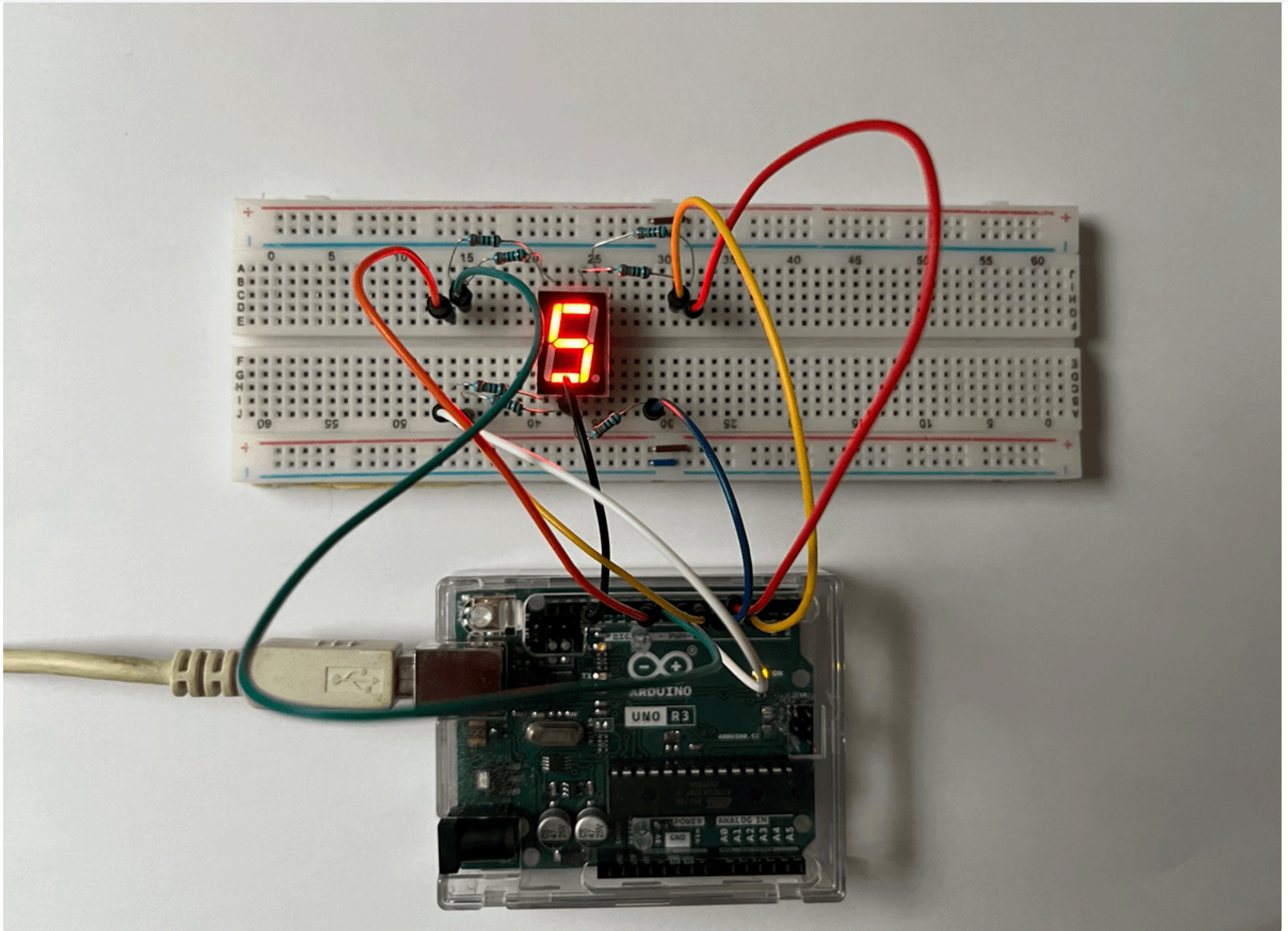
- Le chiffre **4** s'affiche sur notre afficheur 7 segments pendant 2 secondes :





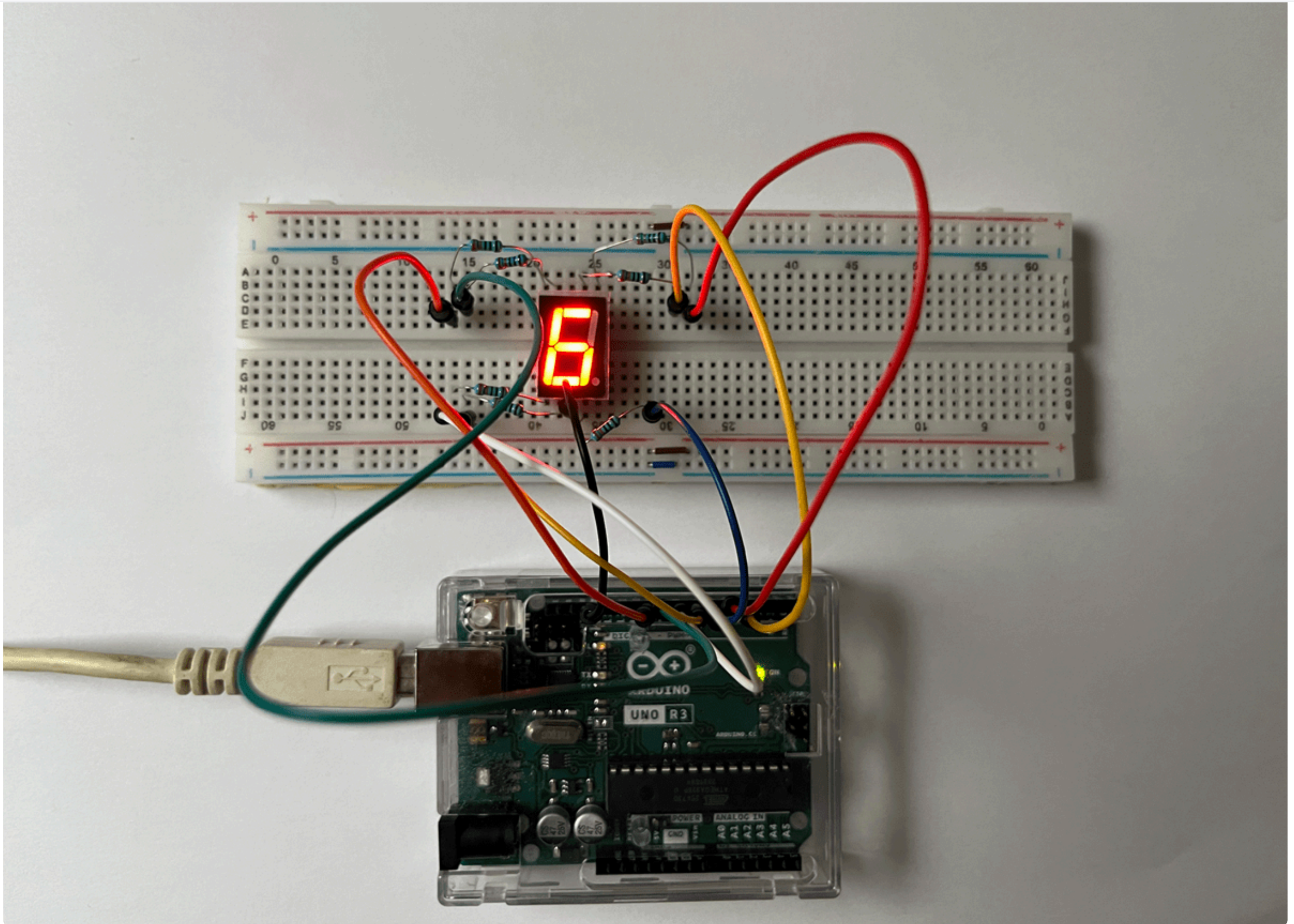
- Le chiffre **5** s'affiche sur notre afficheur 7 segments pendant 2 secondes :



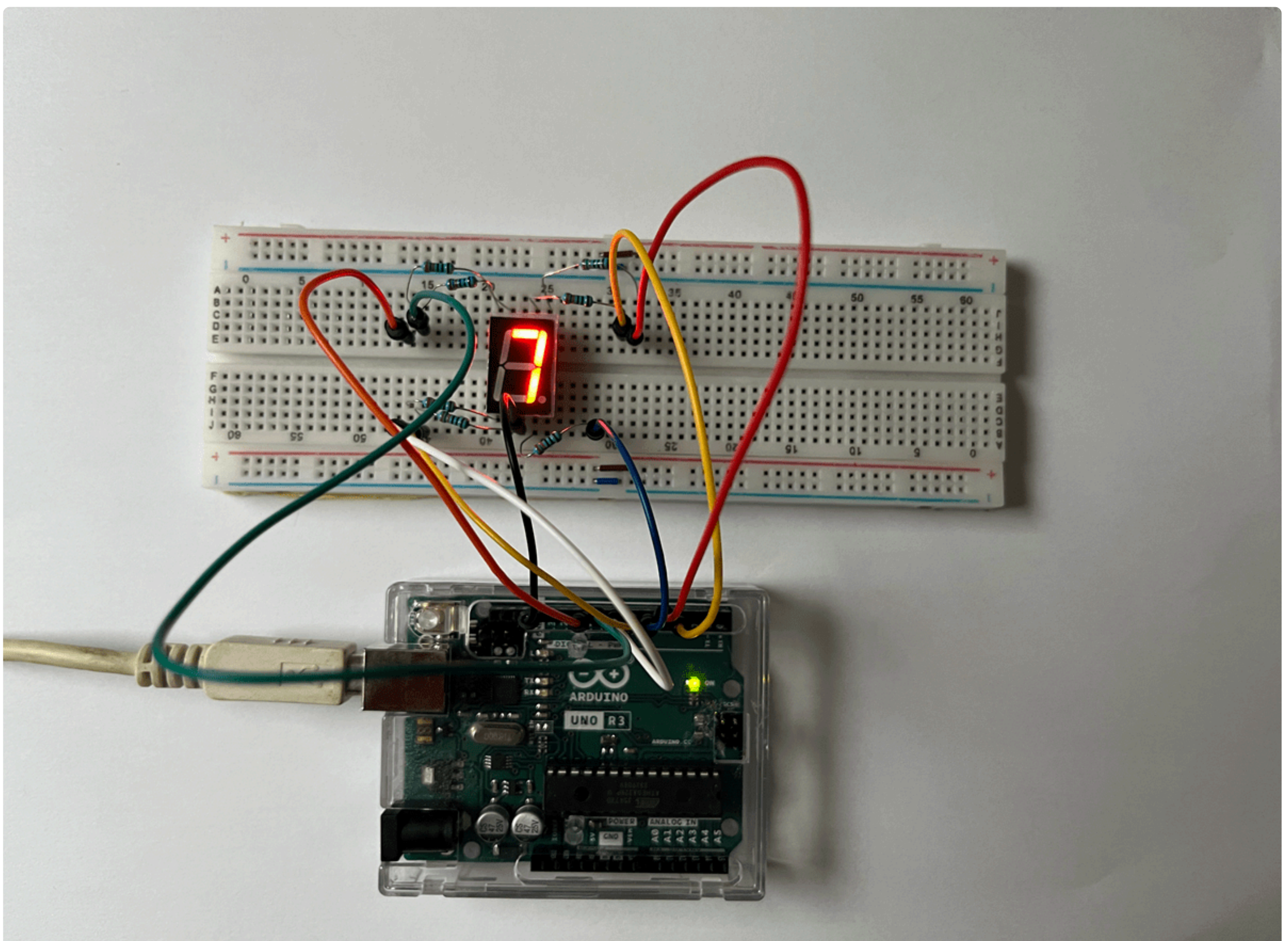


- Le chiffre **6** s'affiche sur notre afficheur 7 segments pendant 2 secondes :

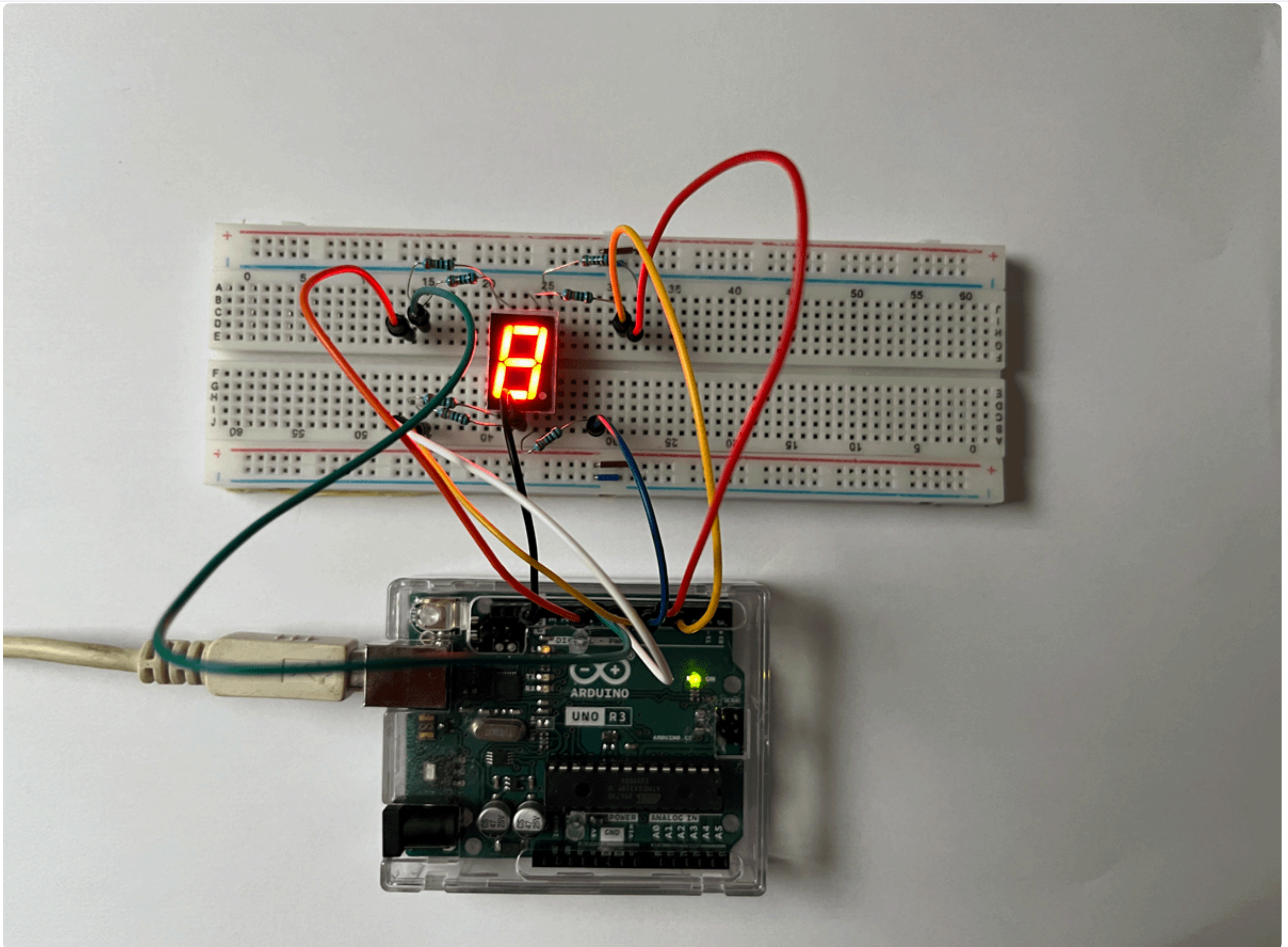




- Le chiffre **7** s'affiche sur notre afficheur 7 segments pendant 2 secondes :

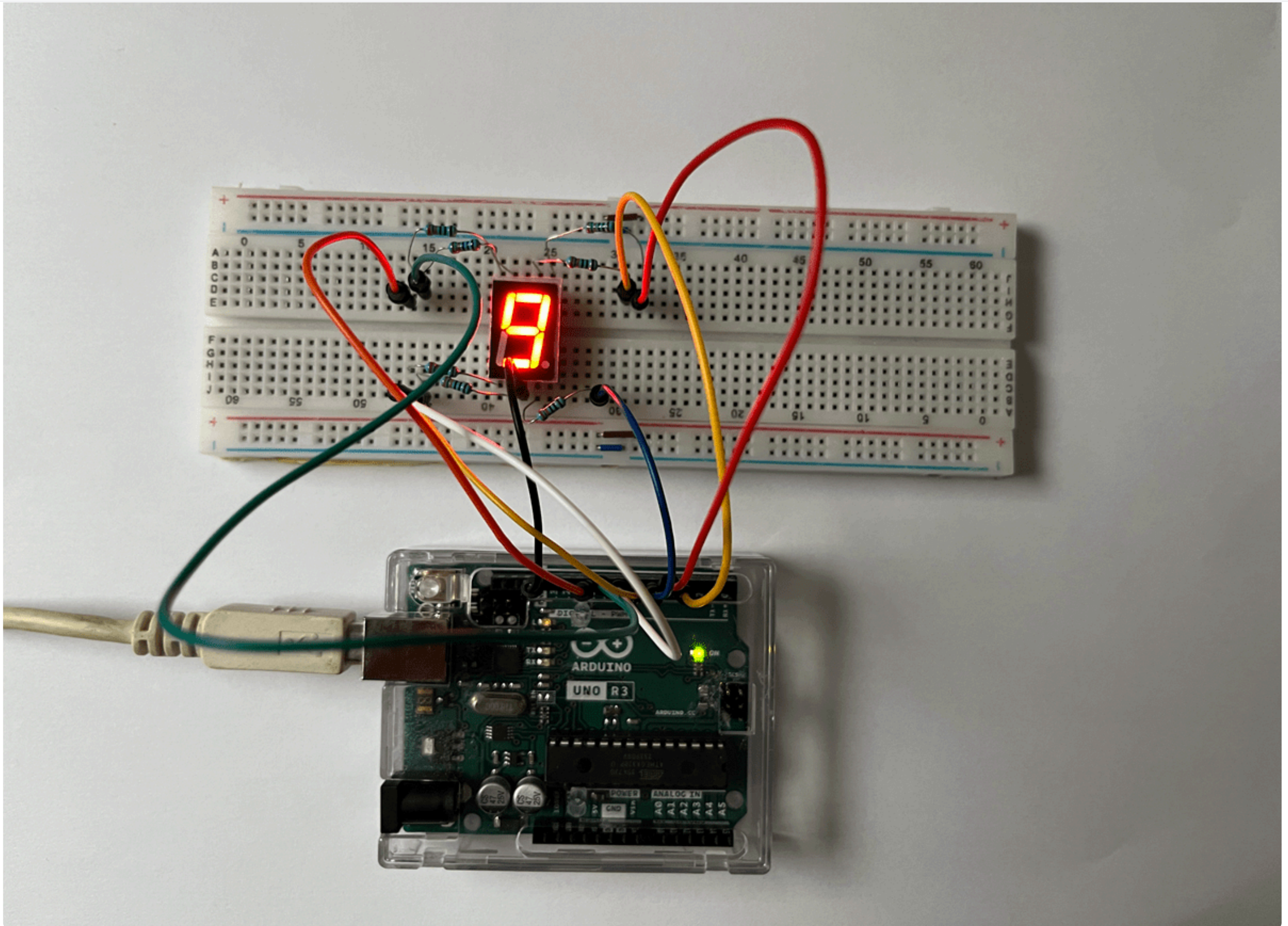






- Le chiffre **9** s'affiche sur notre afficheur 7 segments pendant 2 secondes :





Pour visualiser ce mode de fonctionnement du compteur numérique, nous vous proposons la vidéo ci-dessous le résultat de notre activité proposée :

0:00 / 0:40

